Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 203 (2017) 375–387

Procedia Engineering

www.elsevier.com/locate/procedia

26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain

Symmetry-based decomposition for meshing quasi-axisymmetric components

Flavien Boussuge*, Christopher M Tierney, Trevor T Robinson and Cecil G Armstrong

School of Mechanical and Aerospace Engineering, Queen's University Belfast, BT9 5AH, UK

Abstract

Quasi-axisymmetric structural components are very common in any mechanical equipment containing rotating components, such as turbo machinery. Identifying and exploiting symmetric properties in these components is key to simplify the creation of FE volume meshes. In this paper, a novel approach is proposed to detect exact cyclic symmetries, in order to decompose a quasi-axisymmetric CAD component for FEA. Starting from a B-Rep CAD model, axisymmetric and cyclic repeated sectors are automatically identified to generate a subdivided representation, which can then be used to produce a good quality mesh. Using this new component structure, the resulting decomposition produces a reduced number of mesh-able sub-domains. Symmetry properties can then be inferred to generate the full component mesh from meshes on the individual sub-domains. The approach shows a major reduction of the geometry to be meshed leading to less manual intervention from the user.

Peer-review under responsibility of the scientific committee of the 26th International Meshing Roundtable.

Keywords: Cyclic symmetry, 3D segmentation, B-Rep CAD model, geometric reasoning, hex dominant meshing;

1. Introduction

The exploitation of component geometric properties is key to reducing the complexity of the 3D meshing process for finite element (FE) simulation. Minimizing the number of surfaces and curves used to describe the same geometry makes it easier and quicker to perform any necessary modifications for meshing. Even if the simulation must be performed on the entire component, when the structural component contains symmetry properties, it is helpful to identify and to mesh a part of the model and then to apply symmetry operators to that mesh. In mechanical engineering, turbomachinery is based on components which are classified here as quasi-axisymmetric, where large

* Corresponding author. Tel.: +44(0)-28-9097-4277
E-mail address: fboussuge@qub.ac.uk,

1877-7058
Peer-review under responsibility of the scientific committee of the 26th International Meshing Roundtable.
10.1016/j.proeng.2017.09.812

portions of the geometry are axisymmetric but they also contain certain non-axisymmetric features. For example, the blades of a turbine or compressor are repeated symmetrically around the axis of rotation, or a casing structure may contain struts repeated at regular angular intervals. The analyst might benefit from these cyclic symmetry to run the simulation only on parts of the model or to reduce the meshing process by repeating the cyclic sector. Currently, to obtain a high fidelity 3D hex dominant mesh of a quasi-axisymmetric CAD, the main industrial process still consists of the following manual tasks:

1. remove small details not influencing the simulation
2. identify the repetitive sub-domains
3. decompose the model into repeated sub-domains and non-symmetric sub-domains
4. mesh the individual sub-domains
5. assemble the meshed sub-domains, applying symmetry and repetition operators, to generate the full component mesh.

The above process requires significant effort and skill to decompose the model into the minimum amount of sub-domains to be meshed and repeated. In addition to manual segmentation, understanding the interaction between the cyclic repetitions and the axisymmetric and non-symmetric regions as well as finding the cutting paths to decompose the part is not trivial for the user. Hence, the identification of local symmetries will benefit defeaturing, reducing the repetitions of detail removal as well as for the segmentation operators. This paper describes an automated symmetry-based segmentation approach of a quasi-axisymmetric model for FE analysis.

The rest of the paper is organized as follows. Section 2 discusses related work in CAD segmentation for FE analysis and symmetry detection. Section 3 describes how the axisymmetric properties and cyclic faces are extracted from the B-Rep model. Section 4 presents how to segment the component using the extracted symmetry properties. Finally Section 5 discusses the benefit of the proposed decomposition approach for Hex dominant meshing.

2. Related work

The present study is part of the incremental approach to FE hex-meshing intended to reduce the complexity of hex meshing generation. The authors propose to automatically identify and decompose solid B-Rep models into meshable sub-volumes. Several approaches have been introduced to automatically decompose solid models into sub-volumes directly appropriate for sweep hex meshing. Price et al. [1], then Robinson et al. [2] use the 3D medial object (MO) to identify thin sheet regions (which are large laterally relative to their thickness). The face-pairing approach has been proposed by Rezayat [3] detecting nearly parallel faces of CAD models. Sun et al. [4] recently extend the face-pairing approach to automatically identify thin sheet regions and hex mesh them with a sweep meshing technique. Makem et al. [5] automatically identify long-slender volumes by using local sizing measures. Boussuge et al. [6] extract generative shape processes from B-Rep objects. The extrusion volumes extracted with their corresponding interfaces are then dimensionally reduced to produce full shell meshed FE models. Finally, Liu and Gadh [7] as well as Lu et al. [8] focus directly on the detection of sweepable sub-volumes for hex meshing using edge loops. The presented related work significantly contributes to the automation of hex meshing generation. However, the algorithms mainly focus on global properties and do not exploit higher level shape properties such as symmetry. Identifying symmetry properties, especially on components containing many cyclic repetitions, can considerably reduce the number of faces and edges to analyse before carrying out the segmentation.

Symmetry detection approaches proposed in the literature can be classified in terms of symmetry range: global or local symmetry and exact or approximate symmetry [9]–[11]. Finding symmetry property of 3D models has been widely studied by the computer graphics community. Numerous algorithms have been proposed to find symmetries on mesh, e.g. [12]. In this current paper, our input model is a B-Rep CAD model. Although the mesh-based approaches are computationally efficient, analysing directly the B-Rep model will allow us to work within the same level of accuracy as the CAD modeller geometric tolerance (typically between 10^{-3} and 10^{-5} of unit length). It will also avoid using any sampling process which might result in very large mesh models when a small sampling distance is required to preserve all the mechanical features. A review can be found in [11] comparing the level of accuracy of the different symmetry detection approaches when working with CAD models. To identify local approximate symmetries, Li et al. [13] extracts discrete points from approximate CAD models, i.e. models with large tolerances [14]. Later, instead of working on the whole B-Rep model, Li et al. [15] first builds a Regularity Feature tree (RFT) decomposing the volume into sets of simpler and closed sub-domains. Then the approximate symmetry detection is applied locally to these sub-domains and propagated through the model.

In terms of exact symmetry detection on CAD models, Tate and Jared [9] identify matching edge loops in a B-Rep model to extract global exact reflectional and cyclic symmetry. The criteria to classify loop are loop type, loop area, loop centroid and loop normal. The matching edge loops properties are then used to construct the axes and planes of symmetry. Jiang [10] identifies global exact cyclic and reflectional symmetries using the feature tree of the object. Working with features is efficient, but the feature tree can be lost when transferring CAD models through different CAD systems. Recently, Li et al. [11] identify local and global exact reflective symmetry. Using a divide and conquer approach, local symmetry planes and axis candidates are first initiated on all vertices, edges and faces. Then, the global reflective symmetries are extracted from the propagation of the local symmetry through the object boundary. This method is efficient but not currently available for cyclic symmetry.

The previous methods described above focus on the detection of symmetry. Relatively few researches exploit the detected symmetry for FEA, particularly to simplify the meshing process. Suresh [16] proposes to automatically detect symmetry in 2D cyclic sketches and linearly swept 3D solids. Once the symmetry of the solid has been detected, the focus is on the cell construction, i.e. the repeated sector, which is generated by cutting the 2D sketch using a vector starting from the sketch centroid. Even if the authors demonstrate the value of exploiting symmetry properties for FEA, the approach is limited to a unique class of swept 3D solids generated by an extrusion of a 2D cyclic sketch. Cao and al. [17] proposed to construct optimal symmetric 2D or 3D sector meshes using a symmetry-constraint local Delaunay refinement. The proposed approach improves the quality of the final mesh by optimizing the repeated sector; however, it produces only a tetrahedral mesh and is limited to fully cyclic components.

In this paper, an approach is proposed to detect exact cyclic symmetries, in order to decompose a quasi-axisymmetric CAD component for FEA. The objective is to identify axisymmetric and cyclic repeated sectors which can then be used to reduce the meshing complexity.

3. Cyclic symmetry identification

3.1. Modelling context and hypothesis

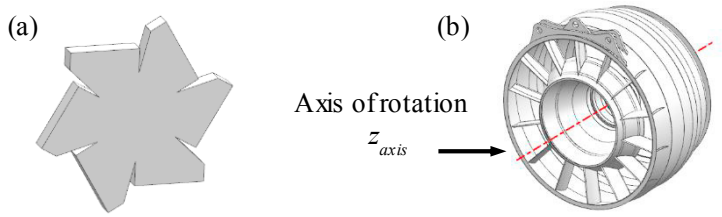


Figure 1 Examples of cyclic components; (a) fully cyclic component created by repeating a sector 6 times; (b) quasi-cyclic component.

In this paper, we focus on quasi-axisymmetric mechanical components. A component is considered as quasi-axisymmetric when it contains a major axis about which most of its geometry is symmetric. Part of the geometry, however, remains non-axisymmetric and cannot be obtained by revolving a 2D sketch around the axis. Figure 1 illustrates a fully cyclic component in (a) and a quasi-axisymmetric component in (b). The main hypothesis of this study is to consider the axis of rotation *z-axis* and *z-location* (*z-axis* passes through this point) as known. The *z-axis* is an input to the axisymmetric and non-axisymmetric detection. Cyclic symmetry (Figure 1 (a)) detection is the main focus within the non-axisymmetric regions. This hypothesis simplifies the process of symmetry detection by identifying the symmetries only around the input axis.

To avoid the side effects of the designer's modelling process, we consider the boundary of the object δM containing only maximal faces and maximal edges defined in [6], [11]. Since multiple features may have been used for the construction of M, there is potentially an infinite number of decompositions of δM that do not change the shape of M. Maximal faces and maximal edges produce an intrinsic and unique boundary decomposition of M. Maximal faces/edges are obtained by repeatedly merging adjacent faces/edges sharing a common edge/vertex under the condition that the faces/edges are surfaces/curves of same type and geometry. In this study, the input is a B-Rep model and the output is a list of patterns, formed by groups of faces circularly repeated around the main axis of symmetry. The proposed method focuses on: identification of axisymmetric and non-axisymmetric faces, identification of cyclic faces and grouping of cyclic faces into patterns.

3.2. Axisymmetric properties and identification

The first step in the process is to identify the faces of the model which are fully axisymmetric. In the context of this work, axisymmetric face classification is different depending on the underlying surface type of the face:
Non-planar faces (i.e. surfaces of revolution, cylindrical, conical, and toroidal surfaces) are classified as axisymmetric if both the axis of rotation and the vector defined between the axis-location and the *z-location* are collinear with the *z-axis*, Figure 2 (a). Geometrically the surface is required to span the entire 360° rotation around the *z-axis*, with edge tangent vectors all orthogonal to the *z-axis* (i.e. *z-axis* is normal to the osculating plane of all boundary edges Figure 2 (a)) and with constant curvature along the circumferential direction. The topological definition of an axisymmetric region is a face bounded by two outer boundary loops, Figure 2 (a). An axisymmetric non-planar face cannot be bounded by any inner loop entities. A loop is a connected set of edges bounding a face. An outer loop is one formed around the periphery of a face, while an inner loop represents a hole in a face.
Planar faces are classified as axisymmetric if they have a face normal parallel with the *z-axis* and all bounding edges have an axis and axis-location collinear with the *z-axis* and tangent vectors orthogonal with the *z-axis* Figure 2 (b). In addition, its topological construct must consist of one outer loop and one inner loop. In the special case where the axisymmetric body is a solid shaft the planar face will not contain an inner loop.

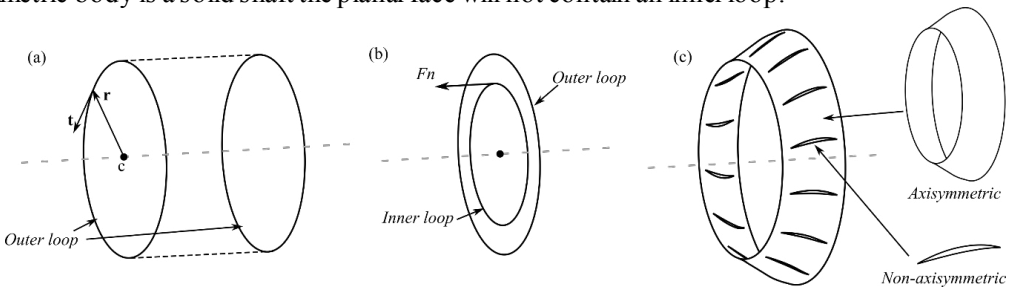


Figure 2 (a) Axisymmetric non-planar face with axis of rotation (dashed grey), axis-location (point c) and tangent (**t**) and radial (**r**) vectors shown, (b) axisymmetric planar face with axis of rotation (dashed grey) and face normal (Fn) shown, (c) Pseudo-axisymmetric face.

If the geometric and topological properties described above are not met then a face can be classified as either non-axisymmetric or pseudo-axisymmetric. A pseudo-axisymmetric face is an axisymmetric face containing inner loops, Figure 2 (c), which can be either non-axisymmetric or cyclic symmetric loops. The outer boundary loops of a pseudo-axisymmetric face must be axisymmetric. Quasi-axisymmetric properties relate to the global properties of a body, whereas pseudo-axisymmetric properties relate to the local properties of faces. Any residual faces are initially classified as non-axisymmetric but are analyzed to check for cyclic symmetry as described in the next sections.

3.3. Identification of cyclic faces

Let us consider a B-Rep object *M* to be analysed. The purpose of this sub-section is the detection of cyclic symmetry in *M*. A cyclic symmetry is defined as a transformation of a shape keeping the shape unchanged by applying a rotation of a given angle $360°/n$ around an axis. In this paper we call the rotation movement *r*, and *n is* the order of rotation symmetry (the number of repetitions). For example, Figure 1 (a) represents a component with a r_6 rotation. The objective is to identify the pattern of cyclic faces $F_{1..p}$ which are repeated circularly in M by a movement r_n around the main axis of rotation.

Table 1 Geometrical conditions to group two faces *F* and *F'*, $\overrightarrow{n_F}$ is the normal vector plane and the axis vector for a cylinder, cone or torus.

Criteria	NURBS	Sphere	Plane	Cylinder	Cone	Torus
Distance	Same distance from the z_{axis}: $dist(\delta F, z_{axis}) = dist(\delta F', z_{axis})$					
Geometric	Same knot vector: $KnotVector(F) = Knot\ Vector(F')$	Same radius	Same scalar projection along z_{axis}: $\lvert\overrightarrow{n_F}.\overrightarrow{z_{axis}}\rvert = \lvert\overrightarrow{n_{F'}}.\overrightarrow{z_{axis}}\rvert$			
	Equivalent control points by rotation: $\forall P_a \in ControlPts(F), \exists P_b \in ControlPts(F'), dist(P_a, z_{axis}) = dist(P_b, z_{axis})$		Same normal orientation	Same radius	Same major radius and half angle	Same major and minor radius

The identification of cyclic faces for a given face F aims at finding all other faces F' which are equivalent to F by a rotation around the *z-axis*. The equivalent faces are grouped together storing the angle of rotation between F and F'. To identify if a face F' is equivalent to F, we use a top-down approach which has three steps. In the first step, we are looking at the geometrical properties of the surfaces F and F'. Step two identifies the topological properties of the respective edges of F and F'. In step three, we compare the underlying curve geometries. Table 1 details the geometrical conditions for two faces F and F' to be grouped together.

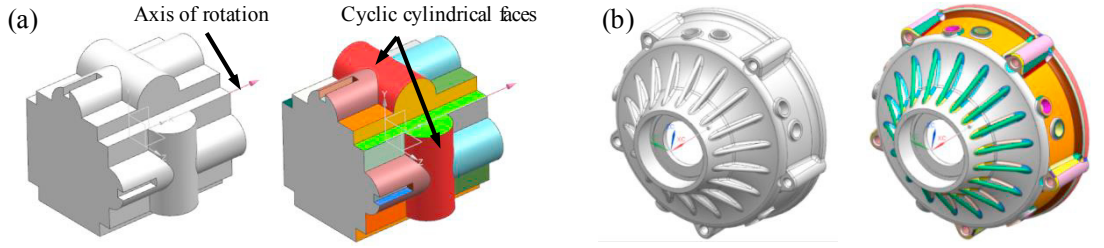
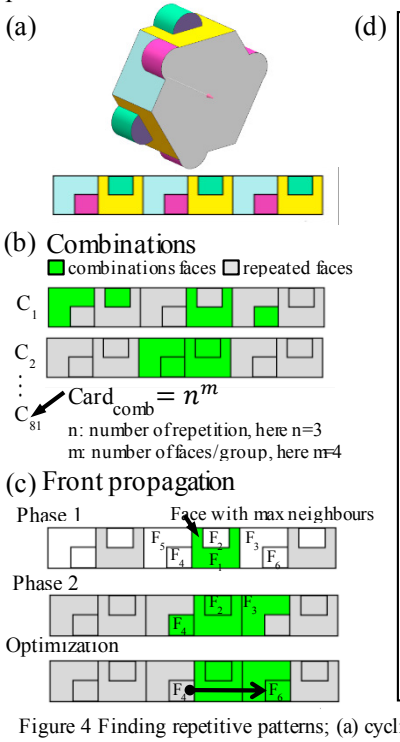


Figure 3 Identification of cyclic faces.

Similarly to the surface comparison of Table 1, the edges and underlying curves of F and F' are compared once the test on the surfaces is verified. The NURBS curves are treated by comparing the knot vectors and the control points. The linear, circular, elliptical curves are compared using their geometric definition. Intersection curves, defined as the intersection between two surfaces, are equivalent if their respective pairs of surfaces are equivalent. Figure 3 shows the cyclic faces, where colour is used to identify the faces which exhibit a given cyclic symmetry.

3.4. Grouping cyclic faces into repetitive patterns

After this step the cyclic faces of object M are organized into groups of cyclic faces G_{n_i} (n is the number of repetitions and i is the group id). The aim is to group the G_{n_i} having the same n and to identify a repeated cyclic pattern.



(d)

Function GroupSameNumberOfRepetition /* group the cyclic faces G_{n_i} having the same n and sharing at least one edge in δM

Result: $GSN[1, \ldots, n]$, $GSN_n \leftarrow [G_{n\_0}, G_{n\_1}, \ldots, G_{n\_m}]$;

Function FindPatterns /* find the patterns of cyclic faces in δM

for each group GSN_n **do**

$F_1 \leftarrow$ face in all the G_{n_i} with the max cardinality (max number of neighbours in GSN_n)

add F_1 to *frontfaces* and *pattern*

while count(*frontfaces*)>0 **do**

frontfaces←GetAdjacentFaces(*frontfaces, pattern*, GSN_n); /*find the compatible faces in GSN_n

visitedfaces←MarkVisitedFaces(*frontfaces*)

add *frontfaces* to *patter;*

add *pattern* to *listofpattern;*

Result: *listofpatterns* [$pattern_1, pattern_2, \ldots, pattern_p$]

Function GetAdjacentFaces /* find the adjacent faces generating the next front

for each face F_i in *frontfaces* **do**

$nei_{Fi} \leftarrow$faces adjacent to F_i;

for each *nei* **in** nei_{Fi} **do**

if *nei* is repeated *n* times **and** *nei* is not in *pattern* **then**

add *nei* to *newfrontfaces*

Result: *newfrontfaces;*

Figure 4 Finding repetitive patterns; (a) cyclic faces, four faces repeated three times every 120°; (b) combinations of faces creating patterns; (c) advancing front method to find a pattern with maximal shared edges between cyclic faces, (d) algorithm grouping cyclic faces into patterns

Patterns are combinations of faces in the G_{n_m} belonging to the same circular repetition r_n. For each G_{n_i}, a combination takes one face of G_{n_i} and adds it to the group pattern P_n. The total number of combinations is n^m (n: number of repetition, m: number of faces in G_{n_i}). Figure 4 (b) illustrates some of the possible combinations derived from the model shown in (a). In order to extract a volume sector, we need to identify a combination forming a repetitive sector where the faces are connected and localised, such as C_3 in Figure 4 (b). The criterion proposed to select a combination of cyclic faces is to maximize the number of shared edges between the faces. A strategy consists in finding all the possible combinations and then selecting the one containing the maximum number of shared edges. However the number of combinations will rapidly increase when the model contains lots of cyclic faces. For example the simple model in Figure 3 (a) already contains $2^{12} = 4096$ combinations. Instead of finding the complete list of combinations, for each group of repetitions, we use a front propagation to find one combination containing connected faces (see algorithm in Figure 4 (d).) and an optimization loop to converge to the list of faces having the maximum number of shared edges.

The front starts with the face having the largest cardinality (sharing most edges with its neighbours having the same n), see F_1 in Phase 1 of Figure 4 (c). This initial face, repeated n time, is added to the group pattern. All the symmetric faces to the initial face are marked and cannot be traversed by the front anymore (faces coloured in grey in Figure 4 (c) phase 1). Then, the front is propagated to an adjacent face (function *GetAdjacentFaces*), if the adjacent face is not symmetric to any of the faces present in the pattern and repeated n times in the solid. Again, all the symmetric faces are marked when an adjacent face is inserted in the pattern. For example in phase 2 of Figure 4 (c), F_2, F_3 and F_4 are selected as compatible adjacent faces of F_1 and all symmetric faces to F_2, F_3 and F_4 are marked. The front ends when all faces of the different groups G_{n_i} are visited or marked. Starting the advancing front with the face having the maximum cardinality allows the algorithm to quickly converge to a solution containing faces already connected. However, this approach cannot ensure that the identified combination will be optimal. As shown in phase 2 of Figure 4 (c), F_4 is found before F_6. Hence, we apply an optimization loop searching iteratively faces in G_{n_i} with a better cardinality regarding the identified combination. For example, in Figure 4 (c), F_4 has been replaced by F_6 as F_6 is sharing two edges with F_3 compared to one edge between F_4 and F_1. Figure 9 (a) illustrate the results of the grouping algorithm obtained on quasi-symmetric symmetric CAD models.

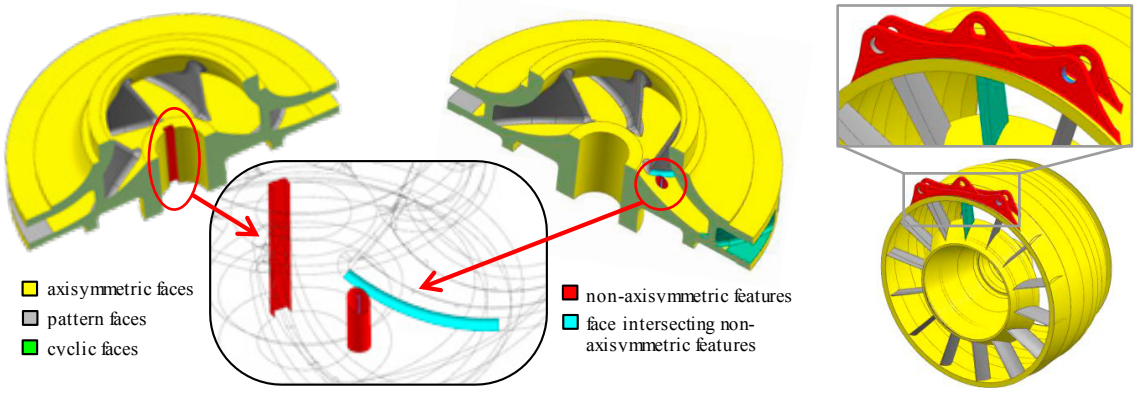


Figure 5 Interaction between cyclic faces and non-axisymmetric features.

3.5. Interaction between groups of symmetry in quasi-cyclic components

The purpose of this paper is to reduce the meshing complexity of quasi-axisymmetric components by identifying axisymmetric regions, non-axisymmetric regions, and any cyclic symmetries in the non-axisymmetric regions. When the model is fully cyclic, the next step is to isolate the repetitive sector (see section 4.1). However, it may happen that the mechanical component contains features which are neither axisymmetric nor exhibit cyclic symmetry. As illustrated in Figure 5, some local features can interfere with the cyclic symmetry. In such a case, one cyclic sector cannot cover the full component but only a portion of it. Then the interaction between the non-axisymmetric faces must be treated separately, Section 4. Other interactions can appear between groups of symmetries when repetitions of faces have a different number of repetitions n. For example in Figure 6, several

groups of symmetry features coexist in the same model, with different numbers of repetitions. Also in this case, a unique sector cannot be found covering the full component.

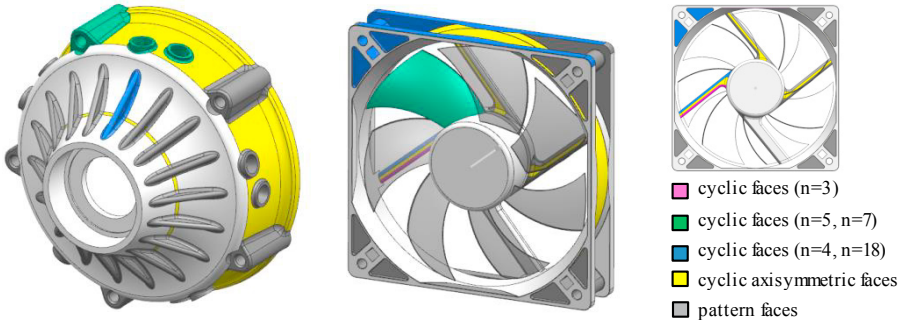


Figure 6 Interaction between groups of symmetries

4. Automatic segmentation for mesh generation

Identifying axisymmetric, cyclic symmetric, pseudo-axisymmetric and non-axisymmetric faces provides the starting point for the segmentation of a cyclic component into a reduced number of mesh-able sub regions, Figure 7. This simplified casing component will provide the use-case for the segmentation.

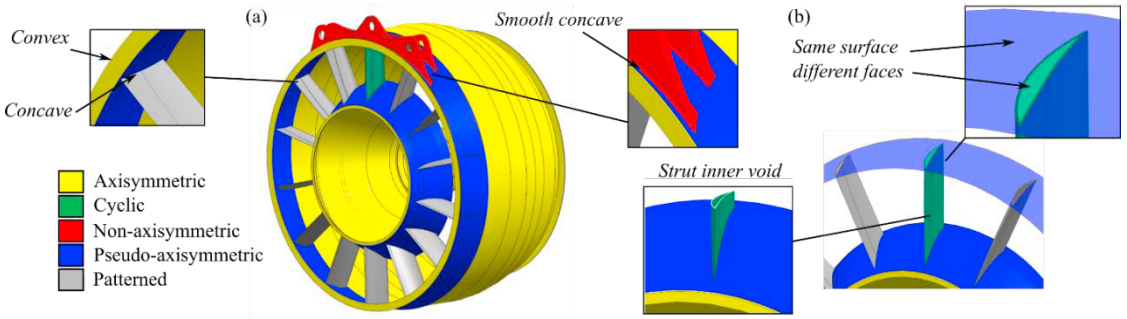


Figure 7 (a) Face classification for casing with certain edge convexity properties shown, (b) Strut definition with inner void.

4.1. Isolation of axisymmetric regions

Axisymmetric region isolation enables the axisymmetric profile to be meshed and swept around the *z-axis* to create a 3D mesh. Isolating an axisymmetric region requires finding appropriate cutting surfaces to partition it from adjacent non-axisymmetric features so it is bounded by all axisymmetric faces. This process begins by interrogating the axisymmetric faces to identify the 'uncommon' boundary (i.e. the edges that bound only one axisymmetric face). Uncommon edges, along with their convexity properties, dictate where a cutting surface must be generated.

The main convexity properties utilised in this work are convex, concave, smooth flat, smooth convex and smooth concave edges [18]. Smooth edge types have bounded faces with parallel normals along the edge, i.e. edges of a concave or convex blend feature, Figure 7 (a). If an uncommon edge is concave then a cutting surface is generated by projecting this edge in the appropriate direction. If the uncommon edge is convex the axisymmetric faces are traversed until a concave edge is located. The projection direction depends on the underlying surface properties of the axisymmetric face it bounds, Figure 8: If the face is planar the projection is opposite to vector **v** in Figure 8, points inside the face and is perpendicular to the edge tangent. If the axisymmetric face is non-planar the edge is projected opposite to the face normal. A cutting surface is created between the source edge and its projection. If the projection face is an axisymmetric face then it is subdivided into two axisymmetric faces. Otherwise, checks are performed to determine if the projected edge intersects with any non-axisymmetric entities. Cutting faces will always be classified as axisymmetric faces. Partitioned volumes are classified as axisymmetric if all bounding faces are axisymmetric, Figure 8 shows the axisymmetric regions automatically isolated from the casing component.

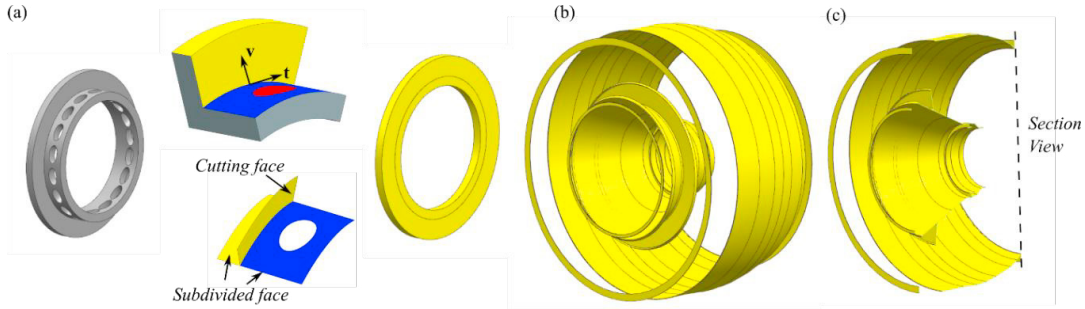


Figure 8 (a) Axisymmetric region cutting plane definition; (b) Axisymmetric regions identified in casing component; (c) Section view of axisymmetric regions in casing component.

4.2. Segmentation of fully cyclic models

When the mechanical structure is fully cyclic, all the groups of cyclic faces G_{n_i} will have the same number of repetitions n. In this case, if the rest of the non-cyclic faces are all axisymmetric, i.e. the face axis / normals are collinear with the z_{axis}, a cyclic sector CS_n can be defined. There is potentially an infinite number of shapes to define CS_n, depending on the shape and location of the cutting surfaces used to divide the object M [16], [17].

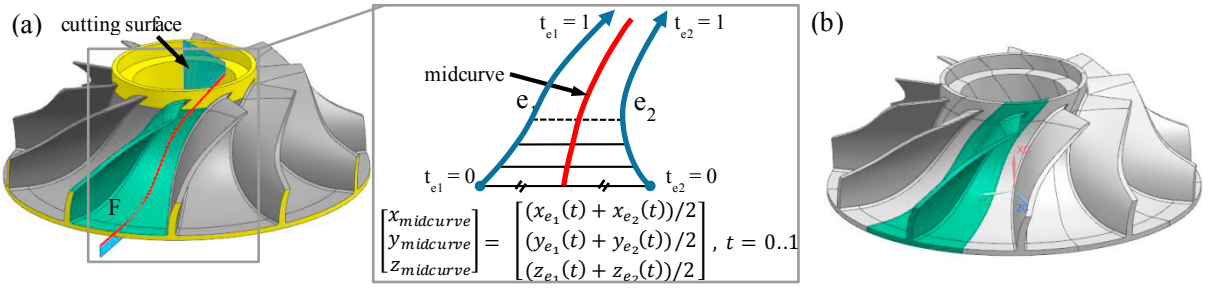


Figure 9 Identification of the cyclic sector on fully cyclic component; (a) in green the group of cyclic faces, in red the mid curves between the non-circular edges extracted from cyclic axisymmetric faces (cylinder, cone, torus, plane or sphere with axis/normal vector/centre collinear with the rotation axis); (b) result of the segmentation using the cutting surface of (a).

Practically, cutting the model in the middle of the axisymmetric cyclic faces, such as F in Figure 9 (a) gives satisfying results. To define the cutting surfaces, we first create axisymmetric midcurves going through axisymmetric faces. These midcurves, such as the red curves in Figure 9 (a), are automatically created by using the parametric definition of non-axisymmetric edges of axisymmetric faces, for example edges e_1 and e_2 of F in Figure 9. To define the cutting surface, the midcurves are projected onto the axisymmetric faces, extended and connected to z_{axis}. Then, the cutting surface is repeated once around z_{axis} to create the second cutting surface. These two surfaces are used to trim the cyclic body M and define the cyclic sector CS_n. Figure 9 (b) shows a cyclic sector CS_{10} in green which can be repeated ten times to obtain the initial model of (a).

4.3. Isolation of cyclic regions in partially cyclic model

Groups of cyclic faces G_{n_i} are interrogated to help identify the partitioning surfaces required to isolate the cyclic sector CS_n associated with each group. As for axisymmetric regions, the uncommon boundary between the faces in a cyclic face group provides the candidate edges where partitioning may occur. If the edge is a concavity, a cutting surface can be directly generated at this location. Alternatively, if the edge is non-concave it's bounded non-cyclic faces are interrogated to determine the best partitioning line. Figure 10 (a) shows two groups of cyclic faces, G_{14_1} (14 repetitions of group id 1) and G_{14_2}, which are the outer and inner groups of cyclic faces bounding the hollow struts connecting the inner and outer structural casing (blue faces Figure 10 (a)). The uncommon boundary of the outer cyclic face group, which contains three faces, returns two loops of concave edges. These are inner loops on the adjacent pseudo-axisymmetric faces of the inner and outer casing.

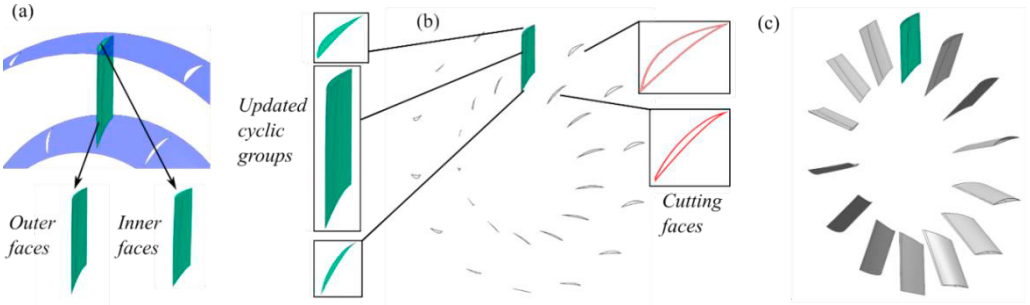


Figure 10 Isolation of cyclic regions (a) Outer and inner cyclic face groups of struts, (b) Cyclic cutting surfaces generated and updated cyclic face groups, (c) partitioned cyclic bodies.

Since the cyclic face group of the inner strut faces contains five connected cyclic faces (the inner void region of the strut is bounded by top and bottom faces) on in Figure 7 (b), there is no uncommon boundary between the face group. Therefore, such cyclic groups classified as inner voids need to be interrogated to determine a connection, if any, with an external cyclic face group. This can be achieved in two ways: **Type 1 connection:** if a face in the cyclic group shares the same underlying surface as a pseudo-axisymmetric face, Figure 7 (b), but does not share a boundary with that face, an extrusion feature is identified and a link between the top and bottom cyclic strut faces and the inner and outer pseudo-axisymmetric casing faces has been determined. **Type 2 connection:** if the first criteria is not met a projection operation is performed between cyclic edges of an inner void cyclic group and another cyclic group to connect cyclic edges in proximity. A cyclic edge must be connected to another cyclic edge.

The connection between two cyclic groups (external and internal) can be extended to all equivalent cyclic faces. The use-case exhibits a type 1 connection allowing a trimmed extraction of the underlying axisymmetric surface (removing inner loops) to be used to isolate the patterned cyclic features (struts). However, if a type 2 connection is present then a best-fit surface is generated between the outer and inner connected cyclic edges and repeated n (14 in this case) times to isolate all patterned cyclic features. Both partitioning approaches generate the same final cutting faces, but the first approach is cheaper as only one cutting tool is required.

Any cutting faces generated, Figure 10 (b), are automatically classified as cyclic faces. Cyclic groups are updated to reflect the partitioning, where faces in cyclic groups that now bound different manifold bodies are separated and new cyclic groups are created to include cyclic cutting faces, Figure 10 (b). Isolated bodies bounded only by cyclic faces are classified as cyclic bodies, which includes the link to the equivalent cyclic patterned bodies, Figure 10 (c).

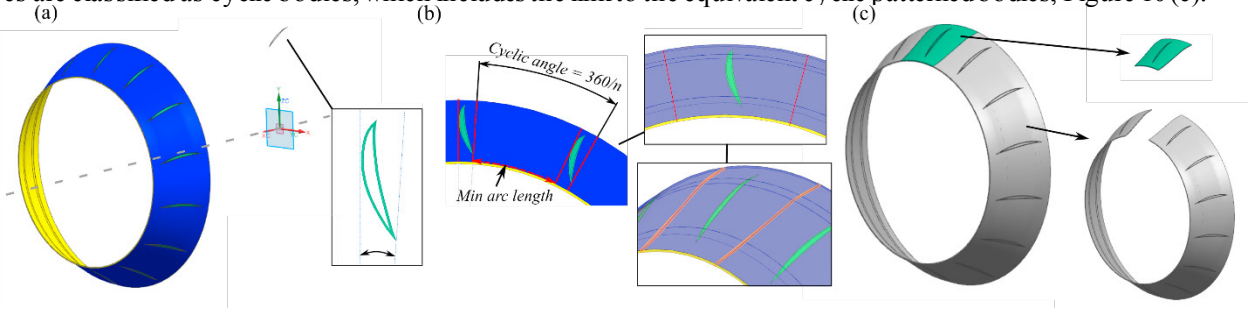


Figure 11 Identifying cyclic symmetry in body bounded by axisymmetric, cyclic symmetric and pseudo-axisymmetric faces: (a) Circumferential span of inner cyclic loop found by projecting to *z-axis plane*; (b) Bounding box (red lines) of inner loop in face identified along with cutting surfaces (inset highlighted); (c) Cyclic sector (green) isolated

4.4. Inferring cyclic properties through cyclic adjacencies

Manifold cyclic cutting faces are linked to one another after partitioning, automatically supplying the adjacencies between the cyclic and non-axisymmetric bodies. Adjacent non-axisymmetric bodies are interrogated to determine if cyclic symmetric sectors can be isolated. If the adjacent body is bounded by all axisymmetric and cyclic symmetric faces then the approach in Section 4.2 is utilised. If the adjacent body is bounded by axisymmetric, cyclic symmetric and pseudo-axisymmetric faces, then the body is entirely cyclic, Figure 11 (a), provided the pseudo-axisymmetric

faces do not contain conflicting cyclic symmetries (left component in Figure 6). For this body type the cyclic cutting surfaces must be generated at the appropriate position on the pseudo-axisymmetric face, Figure 11 (b).

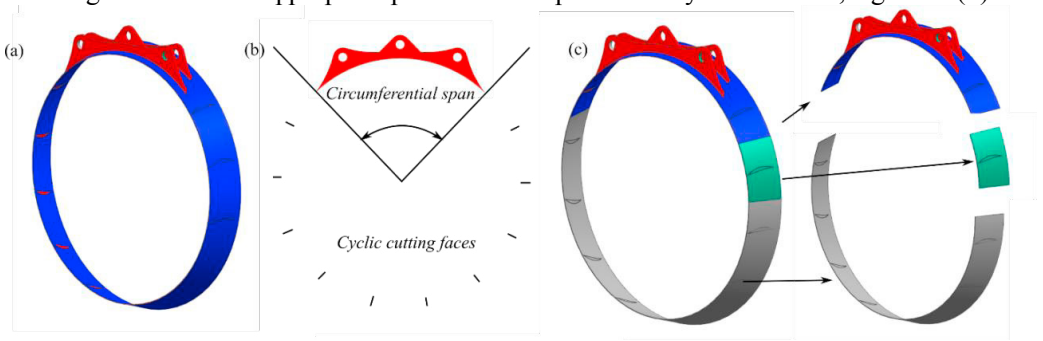


Figure 12 Interaction of non-axisymmetric and cyclic features to identify cyclic symmetry: (a) Non-axisymmetric body; (b) Cyclic faces outside the circumferential span of the non-axisymmetric features; (c) Isolated non-axisymmetric (blue), cyclic (green) and repeatable (grey) bodies.

As the pseudo-axisymmetric face is bounded by cyclic edges, which are inner loops and bound cyclic faces, it is necessary to determine the proximity between these inner cyclic loops. The circumferential span of the cyclic inner loop face is found by projecting its discretised boundary to the *z-axis* plane, Figure 11 (a). The bounding box, inside the face, of the inner loop is generated by extending points generated at the extremities of the circumferential span to the boundary, Figure 11 (a) inset. This bounding box is rotated by the cyclic angle to get the minimum arc length between bounding boxes. If the minimum arc length is six times larger than the target element size (3 elements either side of the partitioning line), a planar cutting face can be generated. Otherwise an approach similar to that described in Section 4.2 must be utilised to find the best cutting surface. The cutting surface is generated by traversing from the mid-point of the minimum arc between the bounding boxes. Traversing from this point to the next axisymmetric edge (pseudo-axisymmetric faces are bounded by two peripheral axisymmetric edges), and so forth, generates a closed loop of curves, Figure 11 (b) red lines inset top, which are used to generate the cutting face. Rotating and copying the cutting face allows the cyclic sector to be isolated, Figure 11 (c). If an adjacent body is bounded by cyclic symmetric and non-axisymmetric faces, Figure 12 (a), it is interrogated to determine if a portion of the body can be isolated as cyclic symmetric, i.e. the cyclic angle is θ/n where $\theta<360°$. In this case the circumferential span of the non-axisymmetric features is considered and is determined by projecting its discretised boundary to the *z-axis* plane, Figure 12 (b). Cyclic cutting faces lying outside, but closest to, the circumferential span of the non-axisymmetric features are used to isolate the non-axisymmetric and cyclic bodies, Figure 12 (c). A cyclic sector within the cyclic region is isolated by rotating a cutting face adjacent to the non-axisymmetric bodies by the cyclic angle, green sector Figure 12 (c).

4.5. Sweep and transition identification

Axisymmetric regions in the model can be meshed by quad meshing the axisymmetric section and sweeping it around the *z-axis*. The remaining bodies are interrogated using the topological algorithms described in [19] to determine if the can be mapped or swept meshed. Bodies with no meshing algorithm are passed to the thin-sheet finding algorithm developed in [4] in order to extract sweepable regions. Removing the aspect ratio constraint described in [4] allows sweepable regions to be identified without requiring them to be thin-sheet regions, Figure 13. Partitioning of the non-axisymmetric body, Figure 13, generates sweepable bodies with good quality hex elements through the thickness and keeps track of the imprints between decomposed bodies. Figure 13 shows a selection of the isolated bodies after this automated thin-sheet decomposition.

Axisymmetric bodies that are adjacent to non-axisymmetric or cyclic bodies are decomposed to insert a transition region between the incompatible regions. Isolating a transition region allows different mesh densities and patterns at its adjacent cells. Transition regions are sweepable through the thickness, as opposed to circumferentially for an axisymmetric region. The cyan regions in Figure 14 (a) and (b) show the transition regions. Using the interface information, which is tracked during the decomposition, an axisymmetric face adjacent to any non-axisymmetric face is a transition face. Propagating from the transition face into the axisymmetric region allows the transition region to be isolated. The propagation distance takes into account the target element size either side of the transition.

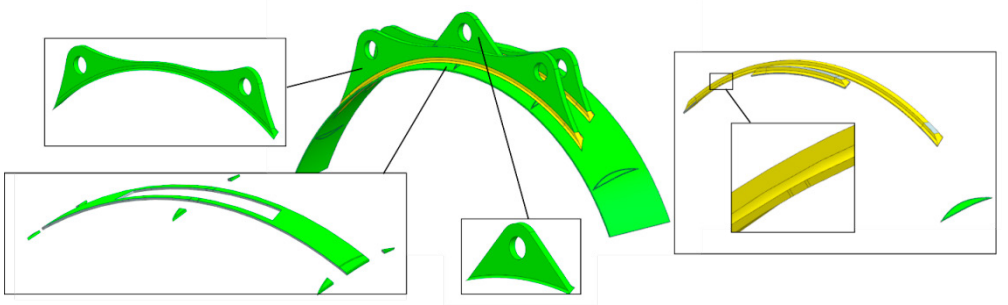


Figure 13 Sweep detection for non-axisymmetric body: sweepable bodies coloured green; imprints of isolated bodies captured (inset right)

5. Results and Discussion

The described approach has been implemented using the C# language and .NET framework APIs in Siemens NX10 [20]. The identification of symmetry properties and the subsequent decomposition has enabled the component in Figure 14 (a) to be decomposed into a number of axisymmetric, cyclic, transition and non-axisymmetric bodies.

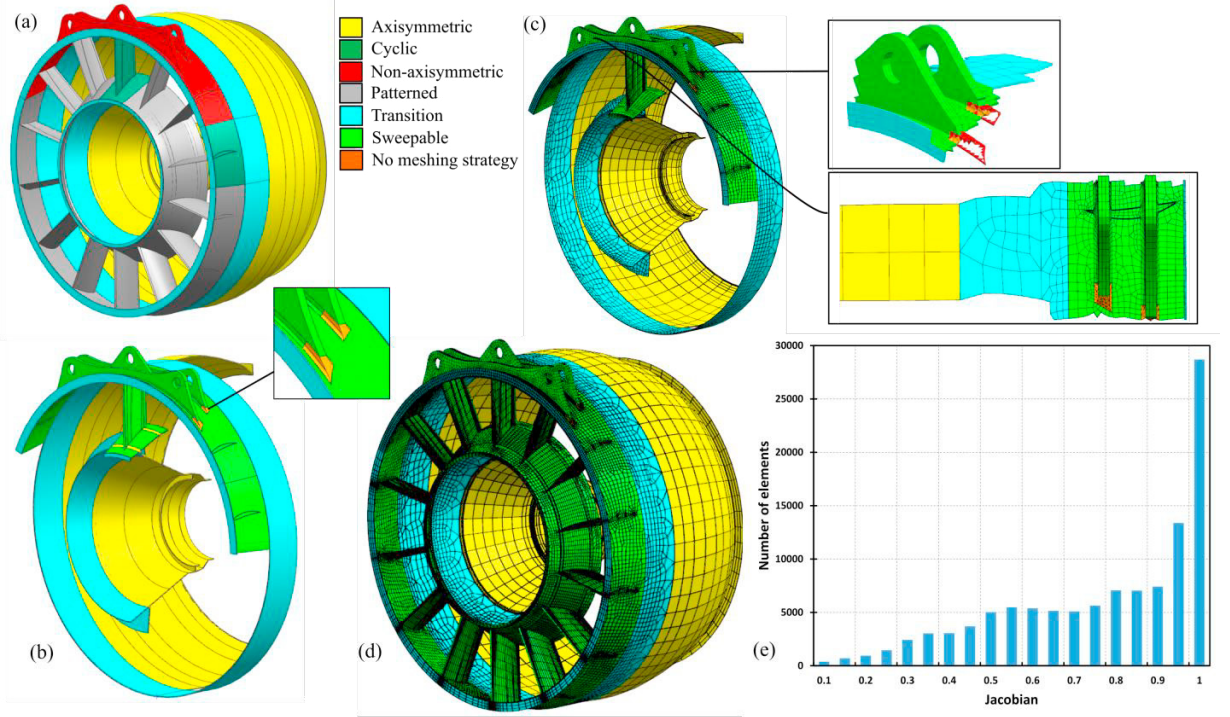


Figure 14 (a) Final decomposition; (b) Minimum mesh-able sub regions (c) Conforming mesh on the minimised decomposition showing the decomposed strut; transition regions and hex-tet transition regions (d) Full component mesh; (e) Jacobian histogram

This automated identification and decomposition, Figure 14 (a), takes 45s (on 64-Bit 3.7GHz Intel Zeon E5-1630 CPU with 32GB RAM). The resulting decomposition reduces the complexity of the hex meshing problem where: 44% by volume of the original model is represented as axisymmetric bodies (yellow); 24% is identified as cyclic (green and grey); 20% as transition regions (cyan) and 12% is non-axisymmetric (red).

Further decomposition is required in order to generate a hex-dominant mesh. The decomposition of the non-axisymmetric body, Figure 13, and the decomposition and imprinting of the cyclic regions adjacent to the strut, is accomplished using the thin-sheet finding algorithm [4], taking 70 seconds. Manual decomposition was required to subdivide the base of the mounts, yellow Figure 13, into sweepable bodies. It was outside the scope of this paper to attempt to automate these manual steps. This results in 97% of the original model being identified as hex mesh-able

and 3% with no meshing strategy assigned. To generate a complete mesh a minimum number of individual sub regions are meshed, Figure 14 (b). Axisymmetric and cyclic symmetry properties used to decompose the model are exploited to generate the full component mesh. Only 29% of the actual original component volume needs to be meshed as axisymmetric regions only require a mesh of their axisymmetric profile to be swept around the *z-axis*.

The minimised mesh, Figure 14 (b), and the full component mesh, Figure 14 (c) are generated manually, where fully automatic meshing is an area of future research. The manual meshing process sheds light on the procedural meshing recipe that would be required in an automated workflow where: axisymmetric regions are meshed with large element sizes in the circumferential direction; sweepable regions are meshed next, prioritising those containing imprinted mappable wall faces; regions with no meshing strategy (orange Figure 14 (b)) are meshed with tet elements with pyramids transitioning, Figure 14 (c) inset highlighted red, with adjacent hex meshed regions; finally transitions regions are swept through the thickness with division number sizing determined from adjacent regions and cyclic patterns, Figure 14 (c). Having previously defined cyclic imprints on adjacent transition regions the full component mesh is generated by repeating the meshes of cyclic bodies and reflecting appropriate transition region meshes. The outcome is a full component mesh which is fully connected, Figure 14 (d) (e) shows the Jacobian histogram for the full component mesh in Figure 14 (d) where 1.8% of the elements have a Jacobian under 0.2.

The benefit of identifying symmetry properties in quasi-axisymmetric components is shown through the automated decomposition described above and the reduction of the meshing burden on the analyst, as only a portion of the model needs to be meshed. Mesh automation needs further research, where edge division number assignment and mesh-geometry parentage associativity for transformed meshes can be informed by the cyclic and axisymmetric properties. Model interfaces and symmetry properties of adjacent regions can also be used to determine meshing recipes depending on the simulation intent of the analysis, where meshing priority can be automatically defined.

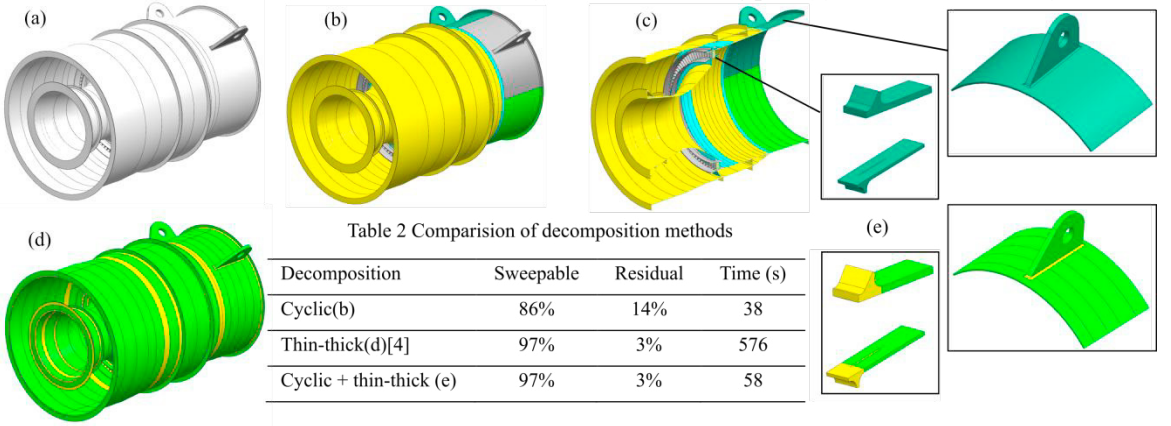


Table 2 Comparision of decomposition methods

Decomposition	Sweepable	Residual	Time (s)
Cyclic(b)	86%	14%	38
Thin-thick(d)[4]	97%	3%	576
Cyclic + thin-thick (e)	97%	3%	58

Figure 15 Use-case 2 (a) CRESCENDO intercase model; (b) Cyclic decomposition; (c) Section view of (b) with non-sweepable cyclic bodies inset; (d) Thin-thick decomposition [4]; (e) Thin-thick of non-sweepable bodies in (c).

To demonstrate the reliability of the algorithm a second use-case is shown in Figure 15 and compared to another decomposition approach for hex mesh generation. The cyclic decomposition described in this paper is shown in Figure 15 (b), where 86% of the original model is identified as sweepable (consisting of axisymmetric, cyclic and transitions) in 38s. The thin-thick approach [4] identifies 97% as sweepable in 576s, Figure 15 (d), which is much longer but the algorithm can be utilised on more generic geometries as opposed to the cyclic decomposition algorithm of this paper. Using the thin-thick approach on cyclic non-sweepable bodies, Figure 15 (c) inset, enables the same total volume (97%) identified as sweepable in only 58s, as opposed to 576s using solely the thin-thick tool. This highlights the benefit of using cheaper algorithms upfront and more expensive algorithms for residual bodies.

Future work around the decomposition involves defining the interactions between cyclic face groups where no obvious subdivision strategy exists. A medial-axis decomposition approach may be required to achieve such decomposition. The symmetry detection described herein is limited to cyclic symmetry. Identifying additional geometric properties such as reflective symmetry or linear repetitions may further reduce the number of sub-regions to be meshed and the burden on the analyst. These additional geometric features would have a direct impact on the meshing recipe required to generate a hex-dominant mesh.

6. Conclusion

This paper presents an approach to partition quasi-axisymmetric CAD components to produce hex mesh-able sub-regions for FE analysis. Taking a B-Rep model and the user-defined axis of rotation, the proposed method initially identifies exact cyclic symmetry. The extracted faces are tagged as axisymmetric, cyclic symmetric, pseudo-axisymmetric and non-axisymmetric faces. These new geometric properties are then used to isolate axisymmetric regions, cyclic sectors and non-cyclic regions. The automatic approach results in the extraction of a minimum of volume sub-regions which can be meshed and repeated to produce the full hex-mesh model.

Acknowledgements

The authors wish to acknowledge the financial support provided by the European Commission via the MUMPS project (704557), a H2020-MSCA-IF-2015 founding scheme and also the support provided by Innovate UK via GEMinIDS (project 113088), a UK Centre for Aerodynamics project. The use-case is a Rolls-Royce model from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 234344 (www.crescendo-fp7.eu). Figures 3, 6 and 9 are created using models provided by the GrabCAD community [21].

References

[1] M. A. Price, C. G. Armstrong, and M. A. Sabin, "Hexahedral mesh generation by medial surface subdivision: Part I. Solids with convex edges," *Int. J. Numer. Methods Eng.*, vol. 38, no. 19, pp. 3335–3359, 1995.

[2] T. T. Robinson, C. G. Armstrong, and R. Fairey, "Automated mixed dimensional modelling from 2D and 3D CAD models," *Finite Elem. Anal. Des.*, vol. 47, no. 2, pp. 151–165, 2011.

[3] M. Rezayat, "Midsurface abstraction from 3D solid models: general theory and applications," *Comput. Des.*, vol. 28, no. 11, pp. 905–915, 1996.

[4] L. Sun, C. M. Tierney, C. G. Armstrong, and T. T. Robinson, "Automatic decomposition of complex thin walled CAD models for hexahedral dominant meshing," *Procedia Eng.*, vol. 163, pp. 225–237, 2016.

[5] J. E. Makem, C. G. Armstrong, and T. T. Robinson, "Automatic decomposition and efficient semi-structured meshing of complex solids," *Eng. Comput.*, vol. 30, no. 3, pp. 345–361, 2014.

[6] F. Boussuge, J.-C. Léon, S. Hahmann, and L. Fine, "Idealized models for FEA derived from generative modeling processes based on extrusion primitives," *Eng. Comput.*, vol. 31, no. 3, pp. 513–527, 2015.

[7] S.-S. Liu and R. Gadh, "Automatic hexahedral mesh generation by recursive convex and swept volume decomposition," in *6th international meshing roundtable, Sandia National Laboratories*, 1997, pp. 217–231.

[8] Y. Lu, R. Gadh, and T. J. Tautges, "Feature based hex meshing methodology: feature recognition and volume decomposition," *Comput. Des.*, vol. 33, no. 3, pp. 221–232, 2001.

[9] S. J. Tate and G. E. M. Jared, "Recognising symmetry in solid models," *Comput. Des.*, vol. 35, no. 7, pp. 673–692, 2003.

[10] J. Jiang, Z. Chen, and K. He, "A feature-based method of rapidly detecting global exact symmetries in CAD models," *Comput. Des.*, vol. 45, no. 8, pp. 1081–1094, 2013.

[11] K. Li, G. Foucault, J.-C. Léon, and M. Trlin, "Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components," *Comput. Des.*, vol. 53, pp. 70–89, 2014.

[12] N. J. Mitra, L. J. Guibas, and M. Pauly, "Partial and approximate symmetry detection for 3D geometry," in *ACM Transactions on Graphics (TOG)*, 2006, vol. 25, no. 3, pp. 560–568.

[13] M. Li, F. C. Langbein, and R. R. Martin, "Detecting approximate incomplete symmetries in discrete point sets," in *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, 2007, pp. 335–340.

[14] B. I. Mills, F. C. Langbein, A. D. Marshall, and R. R. Martin, "Approximate symmetry detection for reverse engineering," in *Proceedings of the sixth ACM symposium on Solid modeling and applications*, 2001, pp. 241–248.

[15] M. Li, F. C. Langbein, and R. R. Martin, "Detecting design intent in approximate CAD models using symmetry," *Comput. Des.*, vol. 42, no. 3, pp. 183–201, 2010.

[16] K. Suresh and A. Sirpotdar, "Automated symmetry exploitation in engineering analysis," *Eng. Comput.*, vol. 21, no. 4, pp. 304–311, 2006.

[17] W. Cao, M. Li, and S. Gao, "Optimal Rotational Symmetry Cell Mesh Construction for FE Analysis by Symmetry-constrained Local Delaunay Refinement," *Comput. Aided. Des. Appl.*, vol. 11, no. 3, pp. 326–334, 2014.

[18] Siemens plm software, "Parasolid." [Online]. Available: https://www.plm.automation.siemens.com/en_us/products/open/parasolid/.

[19] C. M. Tierney, L. Sun, T. T. Robinson, and C. G. Armstrong, "Using virtual topology operations to generate analysis topology," *Comput. Des.*, vol. 85, pp. 154–167, Apr. 2017.

[20] Siemens plm software, "NX." [Online]. Available: http://www.plm.automation.siemens.com/en_us/products/nx/index.shtml.

[21] Stratasys, "GrabCAD." [Online]. Available: https://grabcad.com/library/radial-engine-163, https://grabcad.com/library/centrifugal-working-wheel-1.